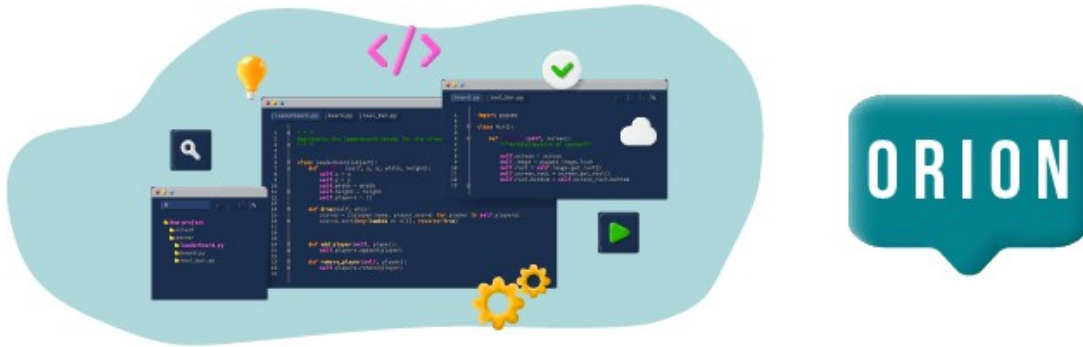


ORION



Justification des choix techniques ***Projet MDD***



Auteur : [Klein Mickaël]
Version 0.0.1

Index

Aperçu / Synthèse.....	3
Choix techniques.....	3
Choix Spring Boot Starter Parent.....	3
Choix Spring Boot Starter Web.....	3
Choix Spring Boot Starter Security.....	4
Choix Spring Boot Starter OAuth2 Resource Server.....	4
Choix Spring Data JPA.....	4
Choix MySQL Connector/J.....	5
Choix Lombok.....	5
Choix Jjwt-API.....	5
Choix Hibernate Validator.....	6
Choix MySQL Database.....	6
Choix Apache Maven.....	6
Choix Angular 17.....	7
Choix Angular CLI.....	7
Choix RxJS.....	7
Choix Jwt-decode.....	8
Choix SOLID.....	8
Choix Factory Pattern.....	8
Choix Git.....	9
Choix GitHub.....	9
Choix techniques non retenus.....	10
Choix Grails.....	10
Choix MongoDB.....	10

Aperçu / Synthèse

Pour répondre aux exigences de l'architecture logicielle et aux contraintes techniques, le choix a été porté sur Typescript/Angular 17 et Java/Spring Boot 3.2.2 comme frameworks principaux pour le front-end et le back-end respectivement. Spring Security a été choisi pour assurer la sécurisation de l'interaction entre le front-end et le back-end, en utilisant l'authentification par JWT. Une base de donnée relationnelle MySQL a été sélectionnée et est un choix cohérent avec les recommandations de l'écosystème Spring. L'architecture client-serveur a été adoptée pour maintenir la séparation entre le front-end et le back-end, conformément aux directives. Les principes SOLID ont été respectés en utilisant notamment Spring Core pour bénéficier de son mécanisme d'Inversion of Control (IoC) et de l'Injection de Dépendances (DI). Les bonnes pratiques recommandées par Angular ont été suivies, et Angular CLI utilisé pour faciliter le développement (respect de l'architecture MVC). Le code a été versionné via Git et Github a été retenu pour la gestion du code source, permettant une collaboration efficace entre les membres de l'équipe de développement et assurant la traçabilité des modifications apportées au projet.

Choix techniques

Choix Spring Boot Starter Parent

choix technique	lien vers le site / la documentation / une ressource	but du choix
Spring Boot Starter Parent	https://spring.io/projects/spring-boot	Développement et maintenance de l'application

•**Justification:** Facilite le développement en fournissant des configurations communes et des dépendances courantes pour les applications Spring Boot.

Choix Spring Boot Starter Web

choix technique	lien vers le site / la documentation / une ressource	but du choix
Spring Boot Starter Web	https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-web	Développement Backend Web API

•**Justification:** Permet la création d'applications web et la gestion des requêtes HTTP. Embarque un serveur Tomcat pour le déploiement.

Choix Spring Boot Starter Security

choix technique	lien vers le site / la documentation / une ressource	but du choix
Spring Boot Starter Security	https://spring.io/projects/spring-security	Sécurisation

•**Justification:** Fournit des fonctionnalités de sécurité robustes telles que l'authentification, l'autorisation et la protection contre les attaques web.

Choix Spring Boot Starter OAuth2 Resource Server

choix technique	lien vers le site / la documentation / une ressource	but du choix
Spring Boot Starter OAuth2 Resource Server	https://docs.spring.io/spring-security/reference/servlet/oauth2/resource-server/index.html	Sécurisation

•**Justification:** Sécurise l'accès aux API REST en exigeant des tokens d'accès valides.

Choix Spring Data JPA

choix technique	lien vers le site / la documentation / une ressource	but du choix
Spring Data JPA	https://spring.io/projects/spring-data-jpa	Gestion des données de la base de donnée

•**Justification:** Permet l'accès et la manipulation de données via JPA (Java Persistence API) et Hibernate, simplifiant l'interaction avec une base de données relationnelle.

Choix MySQL Connector/J

choix technique	lien vers le site / la documentation / une ressource	but du choix
MySQL Connector/J	https://www.mysql.com/products/connector/	Communication avec la base de donnée

•**Justification:** Pilote JDBC permettant la connexion et l'interaction avec une base de données MySQL.

Choix Lombok

choix technique	lien vers le site / la documentation / une ressource	but du choix
Lombok	https://projectlombok.org/	Simplification et maintenabilité du code

•**Justification:** Lombok simplifie la rédaction du code Java en supprimant le boilerplate, ce qui améliore sa maintenabilité.

Choix Jjwt-API

choix technique	lien vers le site / la documentation / une ressource	but du choix
Jjwt-API	https://github.com/jwtk/jjwt	Gestion des tokens

•**Justification:** Bibliothèque permettant de créer, parser et valider des JSON Web Tokens.

Choix Hibernate Validator

choix technique	lien vers le site / la documentation / une ressource	but du choix
Hibernate Validator	https://hibernate.org/validator/	Gestion des données

•**Justification:** Fournit des annotations et des implémentations pour la validation des données au niveau des beans et des types.

Choix MySQL Database

choix technique	lien vers le site / la documentation / une ressource	but du choix
MySQL Database	https://dev.mysql.com/doc/	Base de donnée

•**Justification:** Excellente fiabilité et performances. Bénéficie d'une intégration fluide avec Spring Boot et une documentation exhaustive.

Choix Apache Maven

choix technique	lien vers le site / la documentation / une ressource	but du choix
Apache Maven	https://maven.apache.org/	Gestion des dépendances et build

•**Justification:** Maven simplifie le développement avec Spring Boot en gérant les dépendances et en automatisant la construction du projet grâce à ses fonctionnalités robustes et sa large adoption dans l'écosystème Java.

Choix Angular 17

choix technique	lien vers le site / la documentation / une ressource	but du choix
Angular 17	https://blog.angular.io/introducing-angular-v17-4d7033312e4b	Fonctionnalités du framework

Justification: La version 17 d'Angular apporte plusieurs nouveautés intéressantes en comparaison avec les versions antérieures à la version 16, notamment:

- Control Flow (code plus simple et maintenable)
- Standalone Component (code plus simple et maintenable sans gestion des modules Angular)
- Signals (meilleure réactivité de l'application pour les changements d'états internes)

Choix Angular CLI

choix technique	lien vers le site / la documentation / une ressource	but du choix
CLI Angular	https://angular.io/cli	Développement et maintenance de l'application

•**Justification:** Angular CLI est un outil en ligne de commande qui permet de créer, de construire, de tester et de déployer des applications Angular.

Choix RxJS

choix technique	lien vers le site / la documentation / une ressource	but du choix
Librairie RxJS	https://rxjs.dev	Gestion des données

•**Justification:** RxJS est une bibliothèque de programmation réactive qui offre un ensemble d'outils puissants pour gérer les flux de données asynchrones.

Choix Jwt-decode

choix technique	lien vers le site / la documentation / une ressource	but du choix
Librairie Jwt-decode	https://www.npmjs.com/package/jwt-decode	Sécurité

•**Justification:** Permet de décoder les parties non chiffrées des token JWT afin d'obtenir leur date d'expiration et d'orienter le client en conséquence.

Choix SOLID

choix technique	lien vers le site / la documentation / une ressource	but du choix
SOLID principes	https://www.freecodecamp.org/news/solid-principles-for-better-software-design/	Maintenabilité et scalabilité

•**Justification:** Produire des architectures logicielles plus compréhensibles, flexibles et maintenables.

Choix Factory Pattern

choix technique	lien vers le site / la documentation / une ressource	but du choix
Design Pattern Factory	https://www.baeldung.com/java-factory-pattern	Maintenabilité et scalabilité

•**Justification:** Créer des objets sans exposer la logique de création, réduisant ainsi le couplage et facilitant la maintenance du code.

Choix Git

choix technique	lien vers le site / la documentation / une ressource	but du choix
Git	https://git-scm.com/	Versioning du code

•**Justification:** Assure un versioning du code simple et robuste, largement utilisé dans le monde du développement.

Choix GitHub

choix technique	lien vers le site / la documentation / une ressource	but du choix
GitHub	https://github.com/	Gestion du code source

•**Justification:** Facilite une gestion efficace et universellement adoptée du code source, garantissant sa traçabilité et sa stabilité.

Choix techniques non retenus

Choix Grails

choix technique	lien vers le site / la documentation / une ressource	but du choix
Grails	https://grails.org/	Framework de développement backend

•**Justification:** Spring Boot a été retenu comme framework backend plutôt que Grails en raison de sa popularité et de sa vaste communauté de support, offrant ainsi une documentation bien plus robuste et une maintenabilité future par les équipes de développement simplifiée. De plus, Spring Boot bénéficie d'une large adoption industrielle, ce qui peut faciliter la collaboration et l'intégration avec d'autres outils et services.

Choix MongoDB

choix technique	lien vers le site / la documentation / une ressource	but du choix
MongoDB	https://www.mongodb.com/fr-fr	Base de donnée

•**Justification:** MySQL se révèle plus avantageux que MongoDB pour une application web basée sur des utilisateurs avec des abonnements à des sujets, car sa structure relationnelle permet une modélisation claire et efficace des relations entre les différents partis tandis que MongoDB, en tant que base de données orientée document, peut présenter des défis pour maintenir la cohérence et la transactionnalité dans ce type de système complexe.